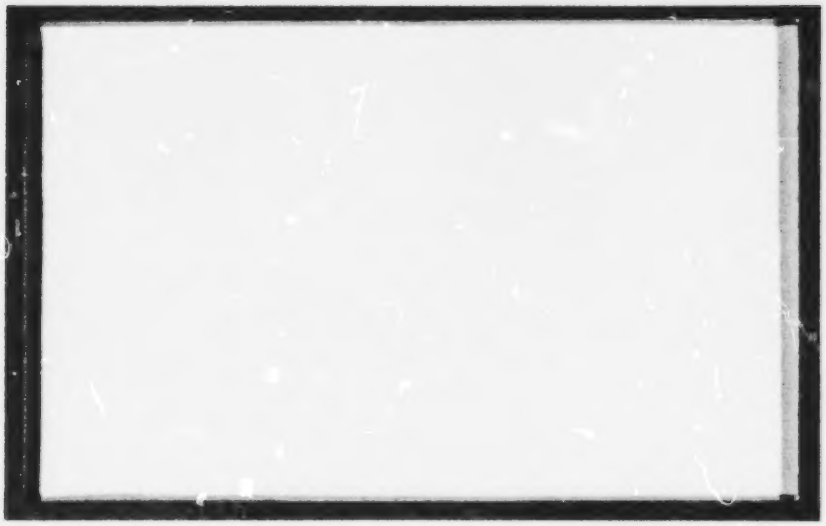
UNIVERSITÉ DE MONTRÉAL



CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

SOLUTION ALGORITHMS FOR NETWORK EQUILIBRIUM MODELS WITH ASYMMETRIC USER COSTS

by

Caroline Fisk and Sang Nguyen

This research was supported in part by a grant from the National Sciences and Engineering Research Council Canada

Université de Montréal

Centre de recherche sur les transports - Publication #167

Avril 1980

Dépôt légal - 2e trimestre 1980

ABSTRACT

We consider algorithms proposed for solving the fixed demand user optimized network equilibrium problem with asymmetric user costs. Because the Jacobian matrix for the costs is asymmetric, the equilibrium conditions cannot be obtained as the Kuhn-Tucker conditions of an optimization problem and alternative solution methods must be sought. Making use of the property that the problem can be expressed in the form of a variational inequality, an algorithm has recently been proposed which has well-defined global convergence properties if the user costs satisfy certain restrictive conditions. This algorithm is compared with some heuristic methods employed in operational models using a small realistic network model in which the costs are particularly asymmetric.

Introduction

In this paper we propose to examine properties of some algorithms designed for solving the fixed demand traffic equilibrium problem with asymmetric user costs for the case where path choice is user optimized. This problem is a generalization of the well known standard model (see Wardrop [18], Beckmann *et al.*[1], Dafermos [4] for example), in that the cost on each network link is considered to be a well-defined function of the flows on all the links. As shown by Dafermos [5] , if the Jacobian matrix for the costs is not symmetric the equilibrium conditions cannot be derived from an optimization problem and convex programming solution techniques are no longer applicable. In operational models of this nature equilibrium traffic patterns have been approximated using heuristic methods. However recently Dafermos [6] has proposed an algorithm which, subject to certain restrictions on the link cost functions, has well defined global convergence properties. The main objective of this paper is to compare empirically the performance of this algorithm with some of the heuristic methods using a small realistic network model in which the link costs have a particularly asymmetric form. The results serve as a basis for evaluating the effectiveness of each method for application on large urban transportation networks.

The paper is organized as follows. In the next two sections we describe the traffic equilibrium problem and discuss the five solution methods incorporated in the study. Then for the purpose of the comparison a small network model is presented for which the link costs have a particularly asymmetric form. Computational results using this network for each of the

algorithms are given and these then serve as the basis for a discussion on the merits and disadvantages of each method.

The user optimized traffic equilibrium problem

The problem is that of determining an equilibrium link flow pattern resulting from the assignment of a fixed O/D travel demand, the route choice criteria being such that at equilibrium no traveller can reduce his cost by unilaterally deciding to change his route. As shown by Smith [16], this property can be expressed concisely by the variational inequality

$$c(f^*)\cdot(f-f^*) \geq 0 \qquad \text{for all } f\in D \tag{1}$$

where $D\subset R^n$ is the closed convex set of feasible link flow patterns,

n is the number of links in the network,

$f^*\in D$ is an equilibrium link flow pattern,

and $c : D\subset R^n \to R^n$ is the link cost vector considered to be a continuous mapping on D.

(Our notation closely follows that of Ortega *et al.* [14]. In particular, all variables are vector variables unless subscripted.)

If the Jacobian matrix $c'(f) = \left[\frac{\partial c_i}{\partial f_j}\right]$ is symmetric on D, f^* can be obtained as a solution of the optimization problem (see Dafermos [5])

$$\min_{f\in D} \int_0^f c\cdot df \tag{2}$$

Furthermore if $c'(f)$ is positive definite (p.d.) for all $f\in D$, f^* is unique.

Smith [16] extended this result to the general case by showing that the solution of (1) is unique if c is strictly monotone on D; i.e. if

$$[c(f) - c(g)] \cdot [f-g] > 0 \qquad \text{for all } f,g \in D. \qquad (3)$$

Also using Brouwer's fixed point theorem and the fact that D is closed and convex, the same author proved the existence of a solution.

Several solution methods with well defined convergence properties for the optimization problem (2) have been developed (e.g. Nguyen [12] and LeBlanc *et al.* [11]), and shown to perform satisfactorily on large scale networks. (See Florian and Nguyen [9] for example). More recently some methods have been proposed for solving the general problem but, with the exception of the algorithm described in [6], convergence properties of these methods are not so well understood. We are now going to present five such methods designated A1 - A5. A1 was proposed recently by Dafermos [6] and is based on a method used for solving variational inequalities. A2 is a modification of A1 suggested in [6] to achieve a better convergence rate, and formulated explicitly in this work. A3 was developed by the authors while A4 and A5 are heuristic algorithms which have already been implemented in operational models.

Solution Algorithms for the user equilibrium problem

In addition to describing the five algorithms incorporated in this study, the following discussion also contains some analysis relating to the convergence properties of some of the algorithms.

A1. The algorithm proposed by Dafermos [6] consists of solving iteratively the user optimized problem with costs given by the affine mapping

$$c(f) = Gf + h^k \qquad (4)$$

where $h^k = \alpha c(f^k) - Gf^k$.

$G \in L(R^n)$ is p.d. and symmetric and ω is a constant. c is assumed to be continuously differentiable on D and uniformly monotone; i.e.

$$[c(f)-c(g)]\cdot[f-g] \geq \alpha|f-g|^2 \qquad \text{for all } f,g \in D; \alpha > 0. \tag{5}$$

(5) implies that α is a lower bound of the minimum eigenvalue of the symmetric part of the Jacobian matrix c' on D. If ν is an upper bound for the maximum eigenvalue of $[c']^t G^{-1} c'$ on D, then for $0 < \omega < \frac{2\alpha}{\nu}$, it is shown in [6] that the sequence f^0, f^1, f^2, ... converges to f^* for any initial $f^0 \in D$. Since (5) is stronger than (3), f^* will be unique.

The assumptions required to obtain the above range for ω are very conservative and even for small networks, it may be difficult to estimate α and ν. Since the value of ω determines the rate of convergence of the algorithm, it would be desirable to ensure that an optimal choice is made. To indicate how the algorithm could be modified to increase the convergence rate, we first prove an alternative convergence theorem to that presented in [6].

Theorem 1. If $\sup_{f \in D} \|I - \omega G^{-1} c'(f)\| < 1$, the mapping T that takes f^k into f^{k+1} is an iterated contraction on D and $\lim_{k \to \infty} f^k = f^*$.

Proof. Let $\Delta f^{k+1} = f^{k+1} - f^k$ and $\Delta c^{k+1} = c(f^{k+1}) - c(f^k)$, then from equation (25) in [6] we have at iteration k+1

$$\|\Delta f^{k+1}\|^2 \leq \{\Delta f^k - \omega G^{-1} \Delta c^k\} \cdot G \Delta f^{k+1} \leq \|\Delta f^k - \omega G^{-1} \Delta c^k\| \|\Delta f^{k+1}\| \tag{6}$$

where $\|x\|$ designates the norm $(x^t G x)^{\frac{1}{2}}$. Let

$$H(f) = f - \omega G^{-1} c(f),$$

then (6) can be expressed as

$$\|\Delta f^{k+1}\| \le \|H(f^k) - H(f^{k-1})\|. \tag{7}$$

Using the mean value theorem this implies that

$$\|\Delta f^{k+1}\| \le \sup_{0\le t\le 1} \|H'(f^{k-1} + t\{f^k - f^{k-1}\})\| \|\Delta f^k\| . \tag{8}$$

$$\text{Let } \sigma(\omega) = \sup_{f\in D} \|H'(f)\| = \sup_{f\in D} \|I - \omega G^{-1} c'(f)\| , \tag{9}$$

then if $\sigma < 1$ it follows from (8) that T is an iterated contraction on D. Since the sequence $f^{k+1} = Tf^k$, k=0,1,2,... remains in D, we have (section 12.3.2 [14])

$$\lim_{k\to\infty} f^k = f^*$$

and the estimate

$$\|f^k - f^*\| \le \frac{\sigma}{1-\sigma} \|f^k - f^{k-1}\| \tag{10}$$

holds.∎

Note that the condition $\sigma < 1$ is comparable to the convergence requirements of the iterative method $x^{k+1} = x^k - \omega G^{-1} c(x^k)$ for solving the nonlinear system $c(x) = 0$.

We now show that it is possible to choose ω such that $\sigma < 1$.

Theorem 2. If $\gamma(f)$ is the smallest eigenvalue of $G^{-1}[c'(f) + c'^t(f)]$ and $\beta(f)$ is the largest eigenvalue of $G^{-1}c'^t(f)G^{-1}c'(f)$ for a given $f\in D$, then $\sigma < 1$ if

$$0 < \omega < \inf_{f\in D} \gamma(f) / \sup_{f\in D} \beta(f) . \tag{11}$$

Proof. Both G and G^{-1} are p.d. and can be expressed respectively by the matrix product R^tR and $R^{-1}[R^{-1}]^t$ where R is an upper triangular matrix. Let

$$y = Rx \quad \text{for any} \quad x \in R^n$$

$$M = I - \omega G^{-1} c'$$

$$N = RMR^{-1}$$

and $$g = \|M\|,$$

then $$g \overset{\text{def}}{=} \sup_{\|x\|=1} \|Mx\| = \sup_{\|y\|_2=1} [y^t N^t N y]^{\frac{1}{2}} = \|N\|_2$$

where $\|\cdot\|_2$ is the Euclidean norm. Thus $g = [\rho(N^tN)]^{\frac{1}{2}}$, where $\rho(N^tN)$ designates the spectral radius of N^tN. Since this matrix is real and p.d., $\rho(N^tN)$ is simply the largest eigenvalue of N^tN. If we substitute for N and use the fact that the matrices $A^{-1}B$ and BA^{-1}, for any $A, B \in L(R^n)$, have the same eigenvalues, $\rho(N^tN)$ will also be the largest eigenvalue of $G^{-1}M^tGM$.

Now $G^{-1}M^tGM = I - \omega P + \omega^2 Q$ where $P = G^{-1}(c' + c'^t)$ and $Q = G^{-1}c'^tG^{-1}c'$. P and Q have real positive eigenvalues and $PQ = QP$; they thus have a common set of eigenvectors. Consequently

$$0 < \rho(N^tN) \leq 1 - \omega\gamma + \omega^2\beta \ .$$

Thus if (11) is satisfied, for any $f \in D$ we have

$$g(f) \leq [1 - \omega \inf_{f\in D} \gamma(f) + \omega^2 \sup_{f\in D} \beta(f)]^{\frac{1}{2}} < 1 \ . \qquad (12)$$

Since $\sigma = \sup_{f\in D} g(f)$, it follows from (12) that $\sigma < 1$. ∎

The condition (11) is quite conservative because in general the smallest eigenvalue of P and the largest eigenvalue of Q will not have the same eigenvector.

Equations (7) and (8) suggest two ways in which ω may be chosen to ensure faster convergence. One approach is to chose ω in iteration k+1 such that the right hand side of (7) is as small as possible. We have

$$\|H(f^k)-H(f^{k-1})\|^2 = \|\Delta f^k - \omega G^{-1}\Delta c^k\|^2 = \|\Delta f^k\|^2 + \omega^2\|G^{-1}\Delta c^k\|^2 - 2\omega(\Delta f^k, G^{-1}\Delta c^k) \quad (13)$$

and this is a minimum with respect to ω when

$$\omega = \frac{(\Delta f^k, G^{-1}\Delta c^k)}{\|G^{-1}\Delta c^k\|^2} \quad (14)$$

If this expression is substituted in (13) it is straightforward to show that $\|\Delta f^{k+1}\| < \|\Delta f^k\|$ so that this method is norm reducing.

The second approach, suggested by (8) but probably more difficult to implement, is to determine ω in iteration k+1 so that

$$\sigma^k(\omega) = \sup_{f\in S_k} \|I - \omega G^{-1}c'(f)\| \quad (15)$$

is minimized, where S_k consists of the points $f = f^{k-1} + t(f^k - f^{k-1})$ with $t \in [0,1]$. Note that theorem 2 ensures that if ω is chosen in this manner in iteration k+1, the method is norm reducing.

A2. In [6] Dafermos points out that if G were chosen to approximate the symmetric part of c' at each step of the iterative process, faster convergence could be expected. This approach would be analogous to the Newton method for solving a system of nonlinear equations, with terms

added to ensure that the linear problem is symmetric. Thus we propose the approximation

$$c^{k+1}(f) = \mu B^k f + c(f^k) - \mu B^k f^k \tag{16}$$

where the constant μ is added for more generality and B^k designates the symmetric part of $c'(f^k)$.

We can obtain bounds for $\|f^{k+1} - f^k\|$ in a manner similar to that used in [6]. Applying (1) at iterations k and k+1 we have

$$[\mu B^k f^{k+1} + c(f^k) - \mu B^k f^k] \cdot [f - f^{k+1}] \geq 0$$

$$[\mu B^{k-1} f^k + c(f^{k-1}) - \mu B^{k-1} f^{k-1}] \cdot [f - f^k] \geq 0$$

Putting $f = f^k$ in the first inequality and $f = f^{k+1}$ in the second, adding the result and simplifying, we find that

$$\|\Delta f^{k+1}\|_k^2 \leq \Delta f^k \cdot B^{k-1} \Delta f^{k+1} - \frac{1}{\mu} \Delta c^k \cdot \Delta f^{k+1} \tag{17}$$

where $\|x\|_k$ designates the norm $[x^t B^k x]^{\frac{1}{2}}$. Repeating the analysis presented in the discussion of Δl, we suggest that μ be determined in each iteration so that at iteration k+1

$$\sigma^k(\mu) = \sup_{f \in S_k} \|[B^k]^{-1} B^{k-1} - \frac{1}{\mu} c'(f)]\|_k \tag{18}$$

is minimized, or more simply chose μ as

$$\mu = \frac{\|[B^k]^{-1} \Delta c^k\|_k^2}{([B^k]^{-1} B^{k-1} \Delta f^k, [B^k]^{-1} \Delta c^k)_k} \tag{19}$$

where $(x,y)_k = x^t B^k y$. Both estimates ensure that $\|\Delta f^{k+1}\|_k < \|\Delta f^k\|_k$.

A3. One problem that can always be solved regardless of symmetry considerations on the costs is the system optimal problem

$$\min_{f \in D} c \cdot f$$

It is straightforward to show that the Kuhn-Tucker conditions for this problem are those of a symmetric user optimized problem with costs $\bar{c} = c + \pi$ where π is the marginal link cost vector: i.e.

$$\pi_a = \sum_b f_b \frac{\partial c_b}{\partial f_a}$$

Thus we propose the following iterative scheme of solving the system optimal problem with costs at iteration k+1 given by

$$c^{k+1}(f) = c(f) - \pi(f^k)$$

and hope that in the limit $k \to \infty$ the link tolls $\pi(f^k)$ are those which transform a system optimal assignment into a user optimal one. This scheme may be viewed as a reverse marginal cost pricing procedure on a transportation network. To ensure uniqueness of the system optimal solution at each iteration, $c \cdot f$ would have to be strictly convex, (or equivalently, $\bar{c}$ must be strictly monotone), in addition to condition (3) which ensures the uniqueness of f^*.

Performing the same operations that produced equation (17) in the previous section, we find that at iteration k+1

$$(\Delta c^{k+1} + \Delta \pi^{k+1}) \cdot \Delta f^{k+1} \leq \Delta \pi^k \cdot \Delta f^{k+1} \qquad (20)$$

where $\Delta \pi^{k+1} = \pi(f^{k+1}) - \pi(f^k)$.

If c is given by the affine mapping

$$c = Af + b ,$$

(20) reduces to

$$2\Delta f^{k+1}\cdot B\Delta f^{k+1} \le \Delta f^{k}\cdot B\Delta f^{k+1} + \Delta f^{k}\cdot C\Delta f^{k+1} \tag{21}$$

where B and C are respectively the symmetric and antisymmetric parts of A. Since by assumption B is p.d. we can define the norm $\|x\| = [x^t Bx]^{\frac{1}{2}}$, and in the special case $C = 0$, we have using the Cauchy-Schwarz inequality

$$2\|\Delta f^{k+1}\| \le \|\Delta f^{k}\| \ .$$

Thus in this case the mapping W that takes f^k into f^{k+1} is an iterated contraction on D and convergence of $f^0, f^1, \ldots$ to f^* is quadratic.

If $C \neq 0$ it follows from (21) that W is an iterated contraction if

$$|\Delta f^{k}\cdot C\Delta f^{k+1}| < \|\Delta f^{k}\|\|\Delta f^{k+1}\| \tag{22}$$

Expressing B as $R^t R$ where R is an upper triangular matrix we find that

$$\begin{aligned} |\Delta f^{k}\cdot C\Delta f^{k+1}| &= |\Delta f^{k}\cdot R^t[R^t]^{-1} C R^{-1} R\Delta f^{k+1}| \\ &\le \|\Delta f^{k}\cdot R^t\|_2 \|[R^t]^{-1} C R^{-1}\|_2 \|R\Delta f^{k+1}\|_2 \qquad (23 \\ &= \|\Delta f^{k}\|\|M\|_2\|\Delta f^{k+1}\| \end{aligned}$$

where $M = [R^t]^{-1} C R^{-1}$. Consequently (22) is satisfied if $\|M\|_2 < 1$, or equivalently if $\rho(MM^t) < 1$. A short computation shows that this reduces to the requirement that

$$\rho(CB^{-1}) < 1 \ . \tag{24}$$

This condition will likely hold for this type of splitting of most p.d. matrices since $|C| \le B$ and the diagonal elements of C are zero. A necessary and sufficient condition for (24) to be satisfied is that there exists a p.d. matrix G such that $H = G - [CB^{-1}]^t G CB^{-1}$ is also p.d. (Stein's theorem). However the authors were not able to find such a matrix.

At the present time a convergence proof for the general case (i.e. for nonlinear costs) could only be formulated under unrealistically restrictive conditions and is not given here.

A4. This method consists of solving in iteration k+1 the standard problem with costs

$$c_a^{k+1}(f) = c_a(f_1^k,\ldots,f_{a-1}^k, f_a, f_{a+1}^k,\ldots,f_n^k) \quad \text{for all } a$$

and setting $f^{k+1} = \bar{f}$, the solution of this problem. Providing that c_a is an increasing function of f_a, other flows being fixed, $\bar{f}$ will be unique.

This method has a parallel in the theory of nonlinear equations, namely the nonlinear Jacobi method and it is possible that convergence for certain classes of cost functions could be proved by taking advantage of this analogy. A4 has been incorporated in the SATURN model [2] and found to perform satisfactorily.

A5. This method is a variant of the incremental loading technique and is described in more detail in [8]. It consists in reallocating flows in each iteration using the scheme

$$F_\ell^k = \begin{cases} F_\ell^{k-1} + \frac{1}{k} \sum_{\substack{m \neq \ell \\ m \in K_i}} F_m^{k-1} & \ell \text{ is shortest path between O/D pair } i \\ F_\ell^{k-1} - \frac{1}{k} F_\ell^{k-1} & \text{otherwise for } \ell \in K_i \end{cases}$$

where F is the path flow vector,

and K_i is the set of paths between O/D pair i.*

* It is been pointed out to the authors that this method is a special case of a more general algorithm proposed for solving the standard problem by J. Almond in Vehicular Traffic Science, Proc. 3rd Int. Symp. on the Theory of Traffic Flow, pages 222-234 (1967).

For the standard problem this is equivalent to applying an optimization procedure with a predetermined steplength $\frac{1}{k}$ in iteration k and a descent direction

$$d_{\ell}^{k} = \begin{cases} \sum_{m \neq \ell,\ m \in K_i} F_m^{k-1} & \ell \text{ is shortest path between O/D pair } i \\ -F_{\ell}^{k-1} & \text{otherwise for } \ell \in K_i \end{cases}$$

as shown in [8]. Convergence proofs for this type of procedure which include the possibility of non-smooth costs have been developed by Polyak (see [15] and references cited).

Of all the methods presented this is the only one which is not nested in structure. It is straightforward to implement (paths do not have to be enumerated since an equivalent link-node formulation exists), and no integrals or derivatives of the cost functions are needed. However experience has shown that the convergence rate is slow [8,15]. Ways of selecting the steplength to ensure faster convergence have been discussed in [15] for the case where the problem has an optimization formulation. An interesting area of research would be to attempt to extend these results to the more general asymmetric problem.

This algorithm is incorporated in the Vancouver CBD model [7] with satisfactory results.

Comparisons of A1 and A2 with affine costs

As a first step in our analysis it was considered instructive to compare A1 and A2 using the network example with affine costs presented in Dafermos [6]. The network is reproduced below in fig. 1 and the travel costs are

$$c_1 = 10f_1 + 5f_4 + 1000$$
$$c_2 = 15f_2 + 5f_5 + 950$$
$$c_3 = 20f_3 + 3000$$
$$c_4 = 20f_4 + 2f_1 + 1000$$
$$c_5 = 25f_5 + f_2 + 1300$$

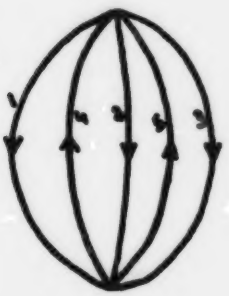


Figure 1

A calculation showed that the optimal value of μ for A2 obtained from (18), is $\mu \sim 1.01$ in which case $\sigma = 0.10884$. The iterative results comparing A1 with A2, using this value for μ, are reproduced in Table 1.

For A1 we have (see [6]) $\|\Delta f^{k+1}\| \leq 0.95\|\Delta f^k\|$, although faster convergence is obtained, while for A2, $\|\Delta f^{k+1}\| \leq 0.10884\|\Delta f^k\|$.

If μ is calculated from (19), the results were found to be almost identical to those in Table 1. Note that since B^k is inchanged throughout the iterative process, the results obtained for A2 are similar to what one would expect A1 to produce with ω calculated from (14) or by minimizing (15).

	Link flow									
	1		2		3		4		5	
k	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2
0	70.0	70.0	70.0	70.0	70.0	70.0	60.0	60.0	60.0	60.0
1	89.4	121.6	76.9	88.4	43.7	0.0	64.6	68.9	56.4	51.2
2	102.4	120.1	82.2	89.9	25.4	0.0	65.9	70.1	54.1	49.1
3	111.2	120.0	86.2	90.0	12.6	0.0	67.3	70.0	56.7	50.0
4	117.2	120.0	89.1	90.0	3.7	0.0	68.2	70.0	51.8	50.0
5	114.8	120.0	90.2	90.0	0.0	0.0	68.8	70.0	51.2	50.0
6	120.0	120.0	90.0	90.0	0.0	0.0	69.2	70.0	50.8	50.0
7	120.1	120.0	89.9	90.0	0.0	0.0	69.5	70.0	50.5	50.0
8	120.1	120.0	89.9	90.0	0.0	0.0	69.7	70.0	50.3	50.0
9	120.1	120.0	89.9	90.0	0.0	0.0	69.9	70.0	50.1	50.0
10	120.0	120.0	90.0	90.0	0.0	0.0	70.0	70.0	50.0	50.0

Table 1. Comparison of A1 and A2 with affine costs

The network model

Broadly speaking two categories of assignment models have emerged which incorporate interactions between conflicting streams of traffic; one in which the link cost functions are obtained explicitly in closed form using traffic flow and queuing theory (e.g. the micro-assignment model developed by Brown *et al.* [3], and the Vancouver CBD model developed by Fisk [7]),and another which generates costs when needed from a simulation model (e.g. the SATURN model developed at the University of Leeds [2]). Both approaches have the common property that a more detailed representation of vehicle movement characteristics is possible, but the first approach has certain advantages from a theoretical and computational point of view. In this approach analytical properties of the cost vector as a mapping of the

link flows can be easily determined, and computationally it is generally cheaper and easier to generate cost values when needed.

Because of the detail incorporated in this class of model, it is particularly suited as a traffic management tool in areas of high traffic density, such as the CBD of a city. While incorporating much of the detail of a simulation model, it has the added capability of being able to determine changes in traffic patterns induced by network modifications. Also, since the cost curves may be derived analytically from vehicle movement characteristics using readily available network data (traffic control information, intersection configurations, etc.), the time consuming fitting of analytical curves (e.g. BPR curves) to travel time data, required for standard models, is not necessary.

For the purposes of comparing the performance of the solution methods introduced above, we use a network (fig. 2) in which interactions between conflicting traffic streams are particularly asymmetric. This situation arises naturally at priority junctions so that in our example a stop sign has been placed at each intersection for one of the approaching traffic streams.

There are 22 nodes, located midblock in our representation; nodes 1 - 4 are traffic origins and 5 - 8 traffic destinations. The O/D matrix used is given in Table 2.

The free flow speed for link a, $c_a(0)$, is calculated using Newton's equations of motion. Let

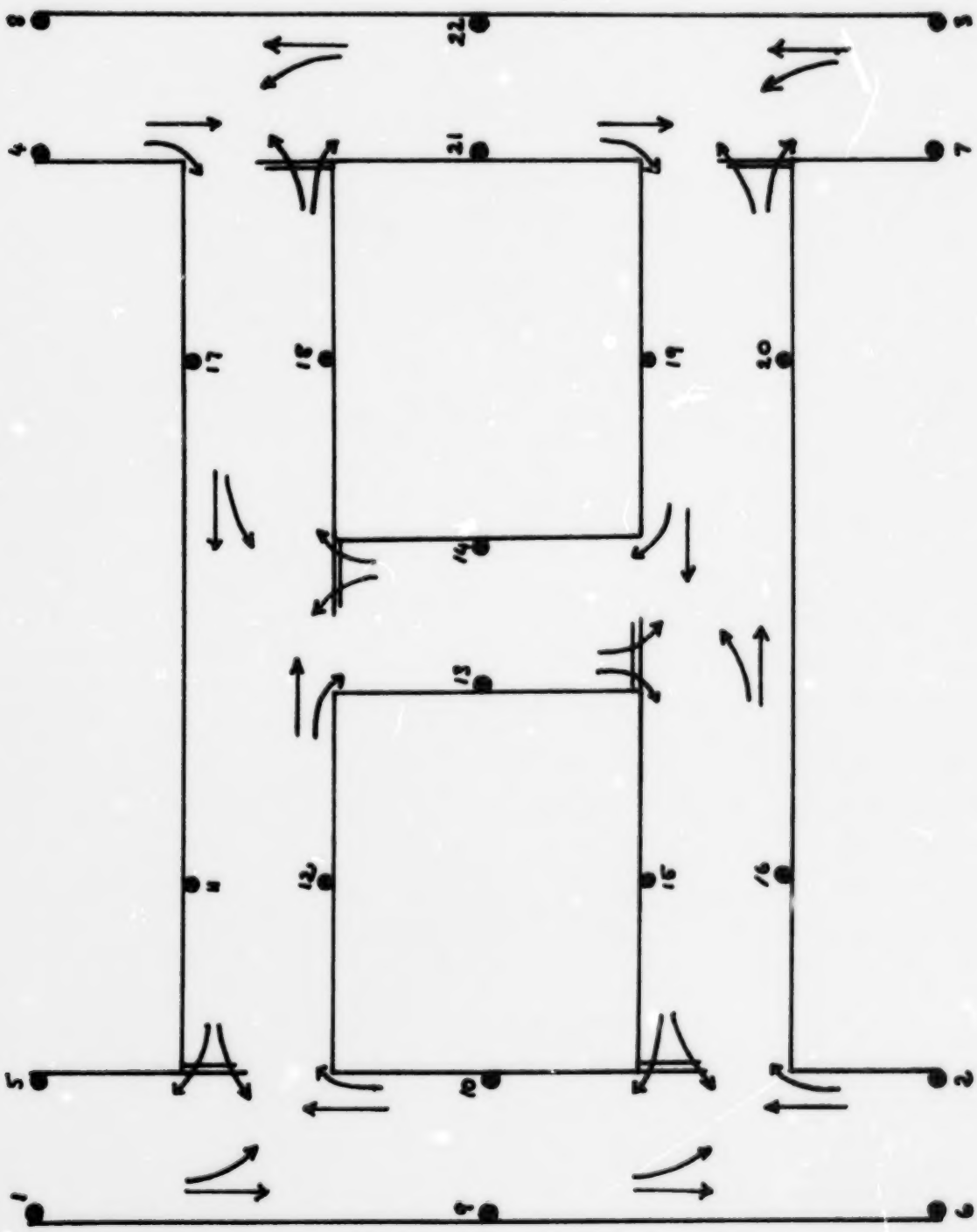


Figure 2

a = vehicle acceleration rate

d = vehicle deceleration rate

ℓ_{1a} - length of approach leg of link a

ℓ_{2a} = length of exit leg of link a

v_{1a} = maximum speed on block face of approach leg of link a

v_{2a} = maximum speed on block face of exit leg of link a

v_c = vehicle cornering speed

and e = time for driver to check for conflicting traffic at stop.

Then

$$c_a(0) = \begin{cases} \dfrac{2(\ell_{1a} + \ell_{2a})}{v_{1a} + v_{2a}} & \text{; unimpeded straight through traffic,} \\ \dfrac{\ell_{1a}}{v_{1a}} + \dfrac{\ell_{2a}}{v_{2a}} + \dfrac{(v_{1a} - v_c)^2}{2dv_{1a}} + \dfrac{(v_c - v_{2a})^2}{2av_{2a}} & \text{; unimpeded turning traffic,} \\ e + \dfrac{\ell_{1a}}{v_{1a}} + \dfrac{\ell_{2a}}{v_{2a}} + \dfrac{v_{1a}}{2d} + \dfrac{v_{2a}}{2a} & \text{stopped traffic.} \end{cases}$$

O \ D	5	6	7	8
1	0	325	300	200
2	400	0	350	250
3	300	200	0	350
4	350	250	400	0

Table 2. O/D matrix used in assignment (veh./hr.)

For simplicity in this model we assume that each turning movement has an exclusive lane so that interactions between traffic on the same intersection approach with different turning movements can be neglected.

Congestion delays on priority links are assumed to have the BPR form:

$$c_a(f) = c_a(0)[1 + \alpha_a (\frac{f_a}{z_a})^{\beta_a}]$$

where z_a is the capacity of link a,

and α_a and β_a are constants.

To calculate congestion delays at stops and for priority traffic turning left against oncoming traffic, we use an expression developed by Gazis *et al.* [10] to determine the expected delay incurred while waiting for a gap in a stream of traffic. For example for a right turn at a stop (fig. 3) this delay is given by

$$b(f_s) = \frac{e^{f_s t} - 1}{f_s} - t$$

where t is the gap acceptance interval.

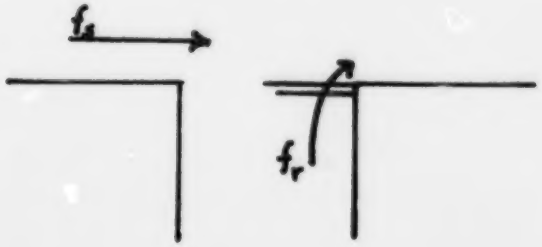


Figure 3

The service rate for the right turning traffic flow f_r, will then be

$$\gamma(f_s) = 1/[e+b(f_s)]$$

and using queuing theory, the expected delay per vehicle for this traffic stream is

$$b(f_s) + f_r/[\gamma^2(f_s) - \gamma(f_s)f_r]. \tag{25}$$

Since we assume that the traffic flow continues indefinitely, (25) approaches ∞ as the arrival rate f_r approaches the service rate $\gamma(f_s)$. To avoid the problem of infinite costs in our model we replaced (25) by the expression

$$b(f_s) + \frac{f_r}{\gamma^2(f_s)} P^m \left(\frac{f_r}{\gamma(f_s)}\right) \tag{26}$$

where $P^m(x)$ is the m^{th} order polynomial approximation for $(1-x)^{-1}$. Graphical comparisons showed that when $m = 50$, (25) and (26) are practically indistinguishable for $f_r < \gamma$; this value for m is incorporated in our model. With this modification, c_a is defined for all $f \epsilon D$ and is continuously differentiable on D.

The total delay for right turning traffic at a stop is thus

$$c_a(f) = c_a(0) + b(f_s) + \frac{f_a}{\gamma^2(f_s)} P^{50} \left(\frac{f_a}{\gamma(f_s)}\right)$$

Similar expressions are used for left turning traffic at a stop and left turning traffic against oncoming traffic.

In this model some second order effects, such as the physical length

of the queue, and delay due to congestion prior to arrival at the end of the stopped queue, have been neglected. Further refinement taking these effects into account is straightforward.

We now indicate some special characteristics inherent in our model and discuss the effectiveness of the algorithms based on our computational experience. For A1 - A4, which require the solution of a standard assignment problem as a subproblem, the program developed by Nguyen *et al.* [13] was used.

Characteristics of network problem

Our network problem has several special characteristics which extend to larger networks and have some bearing on the facility of using the algorithms. For example it is apparent that the cost function c_a for arc a will depend only on flows on links approaching the same intersection. This means that (5) is equivalent to

$$\sum_{a \in \ell_i} [c_a(f) - c_a(g)][f_a - g_a] > 0 \qquad \text{all } f, g \in D, \quad \text{all } i \in I \qquad (27)$$

where I designates the set of intersections in the network, and ℓ_i is the set of links approaching $i \in I$. A similar result has been proved by Smith [17].

Thus the symmetric part of the Jacobian matrix B will have the block structure

$$B = \begin{bmatrix} B^{(1)} & & & \\ & B^{(2)} & & 0 \\ & & \ddots & \\ 0 & & & \\ & & & B^{(m)} \end{bmatrix}$$

where m is the number of intersections and $B^{(i)}$ is the square symmetric matrix with elements

$$B^{(i)}_{k\ell} = \frac{1}{2}\left[\frac{\partial c_k}{\partial f_\ell} + \frac{\partial c_\ell}{\partial f_k}\right] \qquad \text{for } k, \ell \epsilon \ell_i, \; i \epsilon I \; .$$

This facilitates the eigenvalue and matrix inversion calculations required by A1 and A2; the eigenvalues of B are composed of the eigenvalues of each $B^{(i)}$ and B^{-1} is the matrix with blocks $[B^{(i)}]^{-1}$. Since $B^{(i)}$ will in general be sparse and have at most 12 rows for a four-way intersection, these computations could be done in closed form.

We did not set out in this research to contrive a network example satisfying the restrictions on the cost functions required by each algorithm. Rather, part of our interest in this problem lay in testing whether these restrictions would hold for a realistic network model, so to conclude this section we discuss the relevant properties of the mapping c.

c is clearly continuous and differentiable on D and the requirement for A4 that c_a be an increasing function of f_a, other flows being fixed, is satisfied. However we discovered that the monotonicity condition required for A1 and A2 does not hold. This was confirmed by two methods, both of which use the property that the Jacobian of a strictly monotone mapping is p.d. Firstly, we found that the smallest eigenvalue of B is negative for several $f \epsilon D$, and in particular for values of f near an equilibrium solution. Secondly we examined an isolated intersection and showed that (27) does not hold. For example if the links are numerated as shown in figure 4,

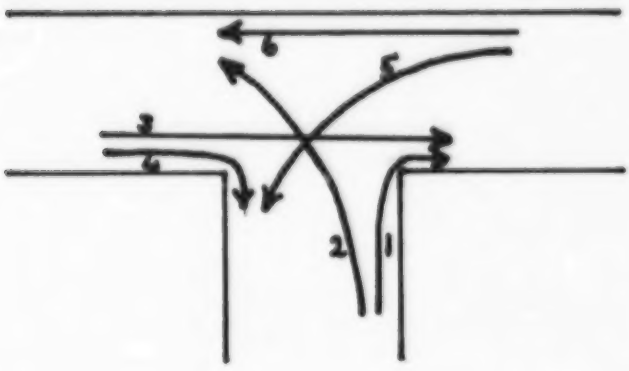


Figure 4

$B^{(i)}$ has the form

$$\begin{bmatrix} B_{11} & 0 & B_{13} & 0 & 0 & 0 \\ 0 & B_{22} & B_{23} & 0 & 0 & B_{26} \\ B_{31} & B_{32} & B_{33} & 0 & B_{35} & 0 \\ 0 & 0 & 0 & B_{44} & 0 & 0 \\ 0 & 0 & B_{53} & 0 & B_{55} & 0 \\ 0 & B_{52} & 0 & 0 & 0 & B_{66} \end{bmatrix}$$

(27) requires this matrix to be p.d. which means that the determinant of each principal minor of $B^{(i)}$ must be positive. However we found that for a wide range of flows, the determinant

$$\Delta = \det \begin{bmatrix} B_{11} & 0 & B_{13} \\ 0 & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

is negative and in fact as $f_3 \to \infty$, other flows being fixed, $\Delta \to -\infty$. This can be explained by the fact that B_{31} and B_{32} are much larger than B_{33}.

Although (3) is not satisfied, the solution of our problem appeared to be unique. The main difficulty arising from the above result is that the parameter α required for A1 is negative, so that the method for estimating ω proposed in [6] cannot be used.

Lastly, we demonstrated experimentally that the requirements for A3 that $c \cdot f$ be strictly convex, appeared to be satisfied. Specifically, let $h(f) = c(f) \cdot f$, then we showed that

$$h(\theta f + (1-\theta)g) < \theta h(f) + (1-\theta)h(g)$$

for a wide range of $f, g \in D$ and $\theta \in (0,1)$.

Performance of algorithms

In this section we compare the experimental convergence rate of the algorithms and mention some problems which arose in implementing them. To give an idea of their convergence properties we compare in Tables 3-7 the total costs $C^P = \sum_i d_i u_i^k$ and $C^a = \sum_a f_a^k c_a(f^k)$ where d_i is the demand between O/D pair i and u_i^k is the cost of the shortest path between O/D pair i in iteration k. We also give the average travel speed at each iteration. At equilibrium to four significant figures $C^P = C^a = 202,4$ (veh.hrs.) and the average speed is 21,04 m.p.h.

Since for a large network the largest portion of CPU time will be spent in computing minimum path trees, we also indicate in the following tables the number of such computations required for a specified number of iterations. In each algorithm the all-or-nothing assignment is used as a

starting point for the iterative process.

A1. Since (5) is not satisfied in our model, ω could not be estimated in the manner suggested in [6]; instead it is computed in each iteration from (14). We found the convergence rate to be extremely slow for all choices of G that were tested. The most effective choice was to set the elements of G as close as possible to those of the Jacobian matrix c', but since the latter may vary considerably during the iterative process, this can only ensure a good convergence rate for a limited number of iterations.

The other major problem with this method resulted from the fact that c^k may be negative. This lead to the existence of negative circuits in the network causing problems in the minimum path computations. One way of overcoming this problem, which we implemented, is to enumerate all elementary paths between each O/D pair but this would not be feasible for large problems.

Iteration	# minimum path computations	C^a (veh.hrs)	C^P (veh.hrs)	Average speed (m.p.h)
1	2	36553.4	14725.1	0.12
2	3	36511.1	14708.4	0.12
3	6	16411.1	6738.8	0.26
5	14	5207.9	2243.9	0.82
10	39	423.4	291.2	9.85
50	150	203.7	201.8	20.91
100	223	202.8	202.1	20.10

Table 3. Computational results for A1

A2. Since B^k is not p.d. in our model $[X^t B^k x]^{\frac{1}{2}}$ does not define a norm. Consequently in our implementation of the algorithm we decided to replace this matrix by the diagonal matrix with elements B^k_{ii}; μ is calculated in each iteration from (19).

This method converged more rapidly than A1. However it also generated negative circuits and the elementary paths had to be enumerated.

Both for A1 and A2 the subproblem with linear costs converged rapidly to the user equilibrium solution, requiring less than 5 iterations of the Frank-Wolfe algorithm to attain an accuracy of 5 significant figures.

Iteration	# minimum path computations	C^a (veh.hrs)	C^P (veh.hrs)	Average speed (m.p.h)
1	2	36595.7	14741.8	0.12
2	4	280.2	232.8	15.20
3	9	199.3	195.5	21.38
5	16	415.1	289.3	10.26
10	37	202.4	202.4	21.05

Table 4. Computational results for A2

A3. As with A1, this method appeared to converge very slowly to the equilibrium solution and in addition, the system optimal subproblem required more iterations than the linear problem of A1 and A2 to reach a satisfactory level of convergence. This method also has the potential for generating negative circuits although this did not occur with our model.

Iteration	# minimum path computations	C^a (veh.hrs)	C^P (veh.hrs)	Average speed (m.p.h.)
1	10	199.1	196.9	21.40
2	20	199.1	197.1	21.40
3	30	199.1	197.3	21.39
5	50	199.3	197.7	21.37
10	100	199.7	198.5	21.33
20	200	200.3	199.5	21.26

Table 5. Computational results for A3

A4. Of all the methods considered, A4 converged the most rapidly requiring only 3 external iterations to approximate the solution to 4 significant figures. This algorithm does not have the problem of generating negative costs but requires the solution of a standard user optimized problem in each iteration which may prove costly for large networks.

Iteration	# minimum path computations	C^a (veh.hrs)	C^P (veh.hrs)	Average speed (m.p.h)
1	5	199.2	197.3	21.38
2	10	202.5	202.4	21.04
3	15	202.4	202.4	21.04
5	25	202.5	202.4	21.04
10	50	202.4	202.4	21.04

Table 6. Computational results for A4

A5. This method converged slowly but like A4 is easy to implement and produced no computational problems. It would appear to have the greatest potential for further development; instead of fixing the steplength at 1/k for iteration k, it may be possible to compute an 'optimal' length in each iteration and to formulate a convergence proof. If the convergence rate could be accelerated it would have considerable advantage over the other methods because it is not nested.

Iteration	# minimum path computations	C^a (veh.hrs)	C^P (veh.hrs)	Average speed (m.p.h)
1	1	36595.7	14741.8	0.12
2	2	5281.8	3317.8	0.81
3	3	236.1	220.8	18.04
5	5	202.4	196.9	18.04
10	10	200.8	198.6	21.22
50	50	201.8	201.5	21.11
100	100	201.8	201.5	21.11
200	200	202.2	202.2	21.04

Table 7. Computational results for A5

Conclusion

Our results tend to indicate that solution methods derived from the theory of variational inequalities may not in the present state of the art be appropriate for solving equilibrium problems for large urban transportation networks. The 3 major difficulties are:

1) The condition of uniform monotonicity on the costs required for a global convergence proof will not be satisfied in a realistic network model.
2. It is evident from the linear example that if ω is computed in each iteration from (14) or (15), one could expect a faster rate of convergence than it is restricted to the interval $(0,\frac{2\alpha}{\nu})$. However in spite of this modification in our application of A1, convergence was very slow.
3. The fact that negative circuits may exist in the network for the linear subproblem complicates the minimum path computations.

The modified A2 algorithm appears to converge at a reasonable rate but it also has the property of generating negative circuits in the linear subproblem.

The fact that condition (3) cannot be expected to hold in a realistic network model opens the door to the possibility of existence of more than one equilibrium solution and the need for algorithms with local rather than global convergence properties. A2 would appear to be suited for this role because of its similarity with the Newton method for solving a system of nonlinear equations.

The present version of A3 is clearly not suitable for use on large networks. It converged extremely slowly in our example and in each iteration

a system optimal problem which also converges slowly must be solved. It is also quite difficult to implement because analytical expressions for the marginal costs for each link must be derived.

A4 and A5 appear to be the most promising algorithms in terms of facility of implementation and cost. It may be possible to develop convergence proofs for A4 for certain classes of cost functions and as mentioned above, A5 has potential for improvement if a way can be found to calculate an optimal steplength in each iteration.

Several variations of the algorithms presented here and their convergence properties will be studied in future research.

REFERENCES

[1] Beckmann, M.J., McGuire, C.B. and Winsten, C.B. (1956). Studies in the Economics of Transportation. Yale University Press, New Haven, Conn.

[2] Bolland, J.D., Hall, M.D. and Van Vliet, D. (1979). SATURN: a model for the evaluation of traffic management schemes. Working paper No. 106, Institute for Transport Studies, University of Leeds.

[3] Brown G. and Scott, R. (1970). Micro-assignment: a new tool for small area planning. Highway Research Record # 322, 149-161.

[4] Dafermos, S.C. and Sparrow, F.T. (1969). The traffic assignment problem for a general network. Journal of Research, National Bureau of Standards - B, 75B No. 2, 91-117.

[5] Dafermos, S.C. (1972). An extended traffic assignment model with applications to two-way traffic. Transportation Science 6, 73-87.

[6] Dafermos, S.C. (1980). Traffic equilibrium and variational inequalities. To appear in Transportation Science.

[7] Fisk, C. (1978). A transportation planning model for detailed traffic analyses. Canadian Journal of Civil Engineering 5, 18-25.

[8] Fisk, C. (1980). Some developments in equilibrium traffic assignment. To appear in Transportation Research.

[9] Florian, M. and Nguyen, S. (1976). Recent experience with equilibrium methods for the study of a congested urban area. In Traffic Equilibrium Methods (Ed. M. Florian), Springer-Verlag, New York.

[10] Gazis, D.C., Newell, G.F., Warren, P. and Weiss, G.H. (1967). The delay problem for crossing an n lane highway. Proc. 3rd International Symposium on Traffic Flow, New York, Elsevier, 267-279.

[11] LeBlanc, L.J., Morlok, E.K. and Pierskalla, W.P. (1975). An efficient approach to solving the road network equilibrium problem. Transportation Research 9 309-318.

[12] Nguyen, S. (1974). An algorithm for the traffic assignment problem. Transportation Science 8 203-216.

[13] Nguyen, S. and James, L. (1975). TRAFFIC - An equilibrium traffic assignment program. Publication #17, Centre de recherche sur les transports, Université de Montréal.

[14] Ortega, J.M. and Rheinboldt, W.C. (1970). Iterative Solution of Nonlinear Equations in Severable Variables. Academic Press, New York.

[15] Polyak, B.T. (1969). Minimization of unsmooth functionals. Zh. Öychisl. Mat. mat. Fiz. 9, # 3, 509-521.

[16] Smith, M.J. (1979). Existence, uniqueness and stability of traffic equilibria. Transportation Research 13B, 295-304.

[17] Smith, M.J. (1980). Junction interactions and monotonicity in traffic assignment. Submitted to Transportation Research.

[18] Wardrop, J.G. (1952). Some theoretical aspects of road traffic research. Proc. Inst. Civil Eng., Part II, 1, 325-378.